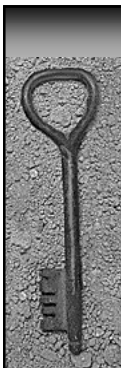


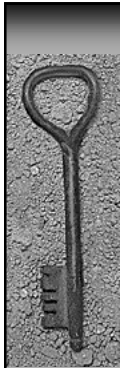
Key Management Workshop

November 1-2, 2001



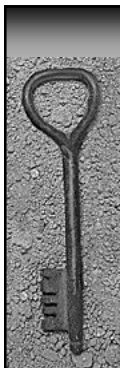
3. Cryptographic Algorithms, Keys, and other Keying Material

- ◆ Approved cryptographic algorithms
- ◆ Security services
- ◆ Keys and other keying material



3.1 Classes of Cryptographic Algorithms

- ◆ Hash algorithms (functions)
 - Un-keyed as one-way collision “free” function
 - Keyed for authentication and integrity (See Symmetric)
- ◆ Symmetric (secret) key algorithms
 - Data confidentiality
 - Part of key establishment
 - Pseudorandom (deterministic) number generators
 - Keyed Hashes for authentication and integrity?
- ◆ Asymmetric (public-private) key algorithms
 - Digital Signatures
 - Key establishment



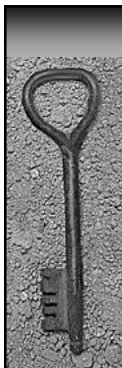
3.2 Cryptographic Algorithm Functionality

- ◆ Many security services are provided using cryptographic algorithms
- ◆ The same algorithm may provide multiple services



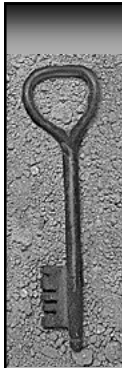
3.2.1 Hash Functions

- ◆ For digital signatures (FIPS 186-2)
- ◆ As part of PSRNG (FIPS 186-2)
- ◆ Keyed MAC (HMAC)
- ◆ Approved hash functions in FIPS 180-2
- ◆ SHA-1, SHA-256, SHA-384, and SHA-512



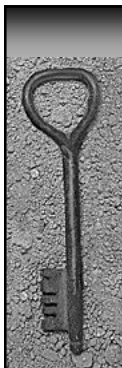
3.2.2 Algorithms for Encryption and Decryption (symmetric key)

- ◆ AES
 - 128, 192, and 256 bit keys
- ◆ Triple DES (TDES)
 - Defined in ANSI X9.52
 - Seven TDES Modes in ANSI X9.52
 - Three distinct keys recommended
- ◆ Modes of Operation
 - NIST Recommendation for four modes plus counter mode



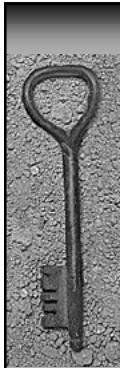
3.2.3 Message Authentication Codes (MACs)

- ◆ MACs using block cipher algorithms
 - NIST recommendation for block cipher MACs (e.g., CBC(AES)-MAC)
- ◆ MACs using hash functions
 - Keyed-hash MAC (HMAC)



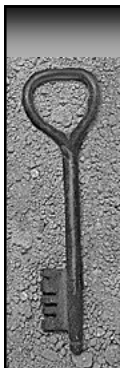
3.2.4 Digital Signature Algorithms

- ◆ DSA (FIPS 186-3)
 - Key sizes between 1024 and 15,360
 - Works with SHA-256, SHA-384, SHA-512
- ◆ RSA Signatures
 - As specified in X9.31 or PKCS #1 v1.5 or higher
 - Key sizes from 1024 in increments of 256 (X9.31)
 - X9.31 and PKCS #1 signature block formats differ
- ◆ ECDSA
 - As specified in X9.62
 - Key size of at least 160 bits
 - Recommended Curves (FIPS 186-2)



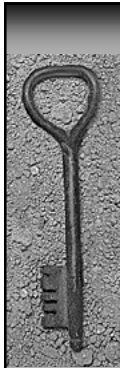
3.2.5 Key Establishment Algorithms

- ◆ Key transport and key agreement schemes to be provide in FIPS XXX
- ◆ Transport may use either symmetric (wrapping) or asymmetric key encrypting keys
- ◆ Agreement uses static and/or ephemeral key pairs to establish a shared secret that is then used to derive the session key(s)
- ◆ FIPS XXX will select schemes from ANSI X9.42, X9.44, and X9.63.



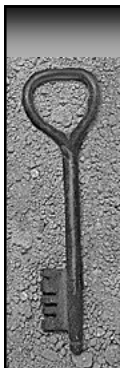
Random Number Generation

- ◆ Needed for generation of keys and IVs
- ◆ Deterministic and non-deterministic
- ◆ Approved deterministic generators from
 - ANSI X9.31
 - ANSI X9.62
 - FIPS 186-2
- ◆ Deterministic often use approved encryption or hash functions and require seed(s)




3.3 Cryptographic Keys and Other Keying Material

- ◆ There are several different classes of keys
- ◆ Many keys are associated with other keying material
- ◆ Each class requires various degrees of protection




Classes of Keys and Protection Requirements

- ◆ Signing keys
 - Private, confidentiality, integrity, assoc with application, assoc with DP and key 2
- Signature verification keys
 - Public, integrity, long term, assoc with application, assoc with entity, assoc with DP and private signing key, validation for association with private key
- Secret authentication keys
 - Secret, confidentiality, integrity, long term, assoc with application, assoc with entity, assoc with authenticated data
- Private authentication keys
 - Private, confidentiality, integrity, assoc with application, assoc with pub authentication key




Classes of Keys and Protection Requirements

- **Public authentication keys**
Public, integrity, long term, assoc with application, assoc with entity, assoc with private authentication key, validation for assoc with private key
- **Long term data encryption keys**
Secret, confidentiality, integrity, long term, assoc with application, assoc with entity, assoc with encrypted data
- **Short term data encryption keys**
Secret, confidentiality, integrity
- **RNG keys**
Secret, confidentiality, integrity, assoc with application




Classes of Keys and Protection Requirements

- ◆ **Key encrypting key used for wrapping**
Secret, confidentiality, integrity, long term, assoc with application, assoc with entity, assoc with encrypted keys
- ◆ **Master key used for key derivation**
Secret, confidentiality, integrity, long term, associated with application, assoc with entity, assoc with derived keys
- ◆ **Keys derived from a Master Key**
Secret, confidentiality, integrity, long term, assoc with application, assoc with entity, assoc with Master Key and protected data
- ◆ **Key transport private keys**
Private, confid, integrity, assoc with appl, assoc with encrypted keys and key transport public key




Classes of Keys and Protection Requirements

- ◆ **Key transport public keys**
Public, integrity, long term, assoc with entity, assoc with key transport private key, validation
- ◆ **Static key agreement private keys**
Private, confidentiality, integrity, long term, assoc with application, assoc with entity, validation of DP and static key agreement public key
- ◆ **Static key agreement public keys**
Public, integrity, long term, assoc with application, assoc with entity, assoc with DP and static key agreement private key, validation
- ◆ **Ephemeral key agreement private keys**
Private, confidentiality, integrity




Classes of Keys and Protection Requirements

- ◆ **Ephemeral key agreement public keys**
Public, integrity, validation
- ◆ **Secret authorization key**
Secret, confidentiality, integrity, assoc with application, association with entity
- ◆ **Private authorization key**
Private, confidentiality, integrity, assoc with application, assoc with public authorization key
- ◆ **Public authorization key**
Public, integrity, assoc with application, associated with entity, associated with private authorization key




Other Keying Material


- ◆ **Domain Parameters**
Public, integrity, long term, assoc with application, assoc with key pair, validation
- ◆ **Initialization Vectors**
Confidentiality, integrity, long term, associated with protected data
- ◆ **Shared Secrets**
Confidentiality, integrity, long term, associated with application, associated with group, associated with other information
- ◆ **Seeds**
Usually require confidentiality
- ◆ **Intermediate Results**
Confidentiality, assoc with application



Classes of Keys

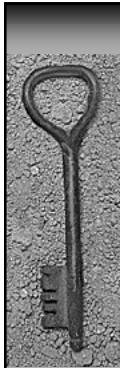
<u>Symmetric</u>	<u>Asymmetric</u>
Secret Authentication Keys	Private Authentication Keys Public Authentication Keys
KEK for Wrapping	Key Transport Private Key Key Transport Public Key
Secret Key Agreement Keys	Static Key Agreement Private Keys Static Key Agreement Public Keys Ephemeral Key Agreement Private Keys Ephemeral Key Agreement Public Keys
Secret Authorization Keys	Private Authorization Keys Public Authorization Keys

	<h2>Classes of Keys</h2>	
	<p><u>Symmetric</u></p> <p>Long Term Data Encryption Keys Short Term Data Encryption Keys</p> <p>ANSI X9.17 Notarized Symmetric Keys</p> <p>Master Keys for Key Derivation Keys Derived form Master Key</p> <p>Random Number Generation Keys</p>	<p><u>Asymmetric</u></p> <p>Signing Keys Signature Verification Keys</p>



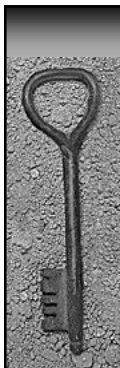
4.4.1 Generation and Distribution of Public/Private Key Pairs

- ◆ Should be generated in Accordance with Approved Standards
- ◆ Generated in FIPS 140-2 Approved module recommended
- ◆ Alternatively generated in access controlled facility
- ◆ Private signing, authentication and authorization keys should not be distributed to other entities



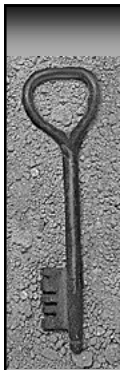
Distribution of Static Public Keys

- ◆ Distribution process should insure that
 - The true owner of the key is known
 - The purpose/usage of the key is known
 - Associated parameters are known
 - The key has been properly generated (e.g., the key validates and/or the owner has the private key)
- ◆ Keys in this category are: signature verification key, public authentication key, key transport public key, static key agreement public key, and public authorization key



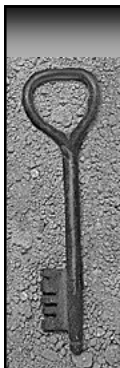
Distribution of Ephemeral Public Keys

- ◆ Ephemeral key pairs are
 - Short-lived
 - Statistically unique to each message
 - Should be generated in accordance with an Approved key establishment scheme (FIPS XXX)
- ◆ Ephemeral public keys should be
 - Distributed using a key establishment process that meets assurances of previous slide
 - Recipient should perform public key validation as specified in FIPS XXX
- ◆ Distribution of Centrally Generated Keys
 - TBD



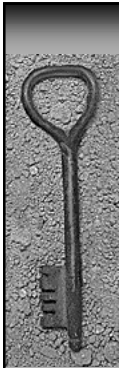
4.4.2 Generation and Distribution of Symmetric Keys

- ◆ Symmetric keys may be
 - Distributed using either a public key or a secret key based key transport system
 - Determined using a key agreement scheme
- ◆ Symmetric keys used for encryption/decryption
 - **Must** be determined by an approved method
 - Should be randomly generated
 - Should be protected consistent with Section 3.3.1



Symmetric Key Generation

- ◆ Symmetric keys should be generated by an Approved key generation algorithm in a FIPS 140-2 module or a controlled access facility
- ◆ Symmetric keys used to protect stored information should be stored in a manner that associates the keys with the information
- ◆ Symmetric keys may be transported to other entities



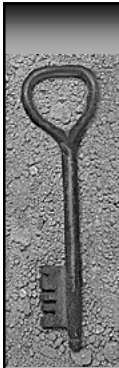
Manual Key Distribution

- ◆ Manually distributed symmetric keys should be either encrypted or use split knowledge
- ◆ The distribution mechanism should assure
 - The authorized distribution of keys
 - That the entity distributing the keys is trusted by both the transmitter and recipient
 - The keys are protected in accordance with Section 3.3.1
 - The keys are received by the authorized recipient
 - The confidentiality and integrity of the keys during transport



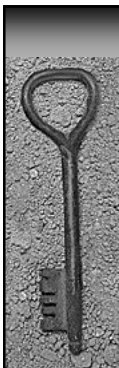
Electronic Key Distribution/Transport of Symmetric Keys

- ◆ Requires other secret or public keys to have been previously distributed
- ◆ Should use FIPS XXX Approved schemes
- ◆ Mechanism should insure that
 - The distributed key is not disclosed or modified
 - The key is protected in accordance with Section 3.3.1
 - The recipient has received the correct key
- ◆ Keys in this category are the secret authentication key, long and short term data encryption keys, key encrypting key for wrapping, master key for key derivation, and secret authorization key.




Key Agreement

- ◆ Approved schemes are provided in FIPS XXX
- ◆ Requires public private key pairs
- ◆ Mechanism should assure that
 - Each entity knows the correct identity of the other entity(ies)
 - The keys in the scheme are correctly associated with the entities
 - The public keys have been validated
 - The derived keys are correct (if key conformation is used)




4.4.3 Generation and Distribution of Other Keying Material

- ◆ Domain Parameters
 - Generated infrequently and shared
 - May be distributed with public keys
 - Should be validated prior to use (FIPS XXX)
- ◆ Initialization Vectors
 - Used with many symmetric modes of operation
 - Protection defined in Section 3.3.2
 - May be distributed with keys or data



4.4.3 Generation and Distribution of Other Keying Material

- ◆ Shared Secrets
 - Computed during key agreement process
 - Used to derive keying material
 - Generated as in FIPS XXX
- ◆ Seeds
 - Initialize deterministic RNG
 - Kept confidential when used to generate keys
 - May be used to generate domain parameters
- ◆ Intermediate Results
 - Occur during computation using cryptographic algorithms
 - Should never be distributed



Questions

